

Explicación técnica del Reto

EJERCICIO 1: FORMULARIO DE CAPTURA DE DATOS

Crear una página llamada **PLA03_Array_Personas.php** con las especificaciones que se enumeran a continuación:

El diagrama muestra un formulario de captura de datos con tres campos de texto etiquetados como 'NIF', 'Nombre' y 'Dirección'. Debajo de estos campos hay un botón etiquetado como 'alta persona'. Abajo del formulario se muestra una tabla con tres columnas: 'NIF', 'Nombre' y 'Dirección'. La primera fila de datos contiene el valor '12345678K' en la columna NIF, 'Peter clemenza' en la columna Nombre y 'Piganerelli avenue, 66' en la columna Dirección. A la derecha de esta fila hay dos botones: 'Baja' y 'Modificar'.

NIF	Nombre	Dirección	
12345678K	Peter clemenza	Piganerelli avenue, 66	Baja Modificar

1.- Características del formulario

El formulario se entrega ya confeccionado dentro del fichero **PLA3_recursos.zip** pero podéis confeccionar el vuestro propio

- Control de tipo **text** para captura del nif
- Control de tipo **text** para captura de nombre y apellidos
- Control de tipo **text** para captura de la dirección
- Control de tipo **submit** para el envío del formulario al servidor
- Una etiqueta **** o similar, situada al lado del botón de alta, para mostrar los mensajes de las operativas (tanto los mensajes informativos como los mensajes de error)
- Una tabla situada debajo del formulario para mostrar los datos de las personas que guardamos en el array. En esta tabla añadiremos los botones de baja y modificación de las persona de cada una de las filas

NOTA: En este PLA vamos a utilizar el mismo fichero donde capturamos los datos para realizar toda la operativa. Para ello en el atributo **action** de la etiqueta **form** indicamos #: **<form method="post" action="#">**

EJERCICIO 2: CREACIÓN DEL ARRAY Y DE LA VARIABLE DE SESIÓN

Las personas que vayamos dando de alta en el formulario las guardaremos en un array que conservaremos en una variable de sesión. De esta forma, cada vez que recarguemos la

página no perderemos los datos que hayamos introducido o modificado con anterioridad.

- Los datos de las personas que vayamos dando de alta los guardaremos en una variable de sesión:

```
$_SESSION['personas']
```

- Para evitar trabajar durante todo el ejercicio con la variable de sesión se recomienda lo siguiente:

- Creamos un array `$arrayPersonas` vacío al inicio del código
- A continuación comprobamos si existe la variable de sesión y, de ser así, copiamos su contenido al `array` vacío

```
if (isset($_SESSION['personas'])) {  
    $arrayPersonas = $_SESSION['personas'];  
}
```

o bien utilizando el nuevo operador de fusión:

```
$arrayPersonas = $_SESSION['personas'] ?? array();
```

- Al final del código php (después de las operativas que añadiremos de alta, baja, consulta y modificación) copiaremos el contenido del array en la variable de sesión

```
$_SESSION['personas'] = $arrayPersonas;
```

De esta forma si, más adelante, cambiamos el destino del almacén de los datos (para guardarlos en un fichero por ejemplo), solo tendremos que modificar el inicio y final del código

EJERCICIO 3: ALTA DE PERSONA

Para la operativa de alta utilizaremos el formulario descrito anteriormente y añadiremos las siguientes acciones:

NIF	22345678J
Nombre	Virgil Solozzo
Dirección	Foscarelli avenue, 45
<input type="button" value="alta persona"/> Alta efectuada	

- Tendremos que detectar con la instrucción `isset($_POST['alta'])` cuando el usuario pulsa el botón de 'Alta persona' (ver la descripción del PLA anterior si hay alguna duda de como se hace esta operativa)

- Recuperar los datos del formulario (utilizando `$_POST` o `filter_input()` indistintamente) y suprimir los espacios en blanco que puedan contener al principio y al final. Ejemplo, si el usuario introduce:

`___Peter clemenza___` (donde cada `_` es un espacio en blanco)

Deberíamos obtener después de recuperar el dato:

`Peter clemenza`

NOTA: para suprimir espacios podemos utilizar la función `trim()`. Ejemplo: `$nif = trim($_POST['nif']);`

- Validaremos que todos los datos sean obligatorios. Para ello utilizaremos un bloque `try...catch` donde incluiremos tanto las validaciones como el código que dependa de ellas. En caso de error mostraremos un mensaje en el `span` que hay al lado del botón de alta persona. Ejemplo:



- En caso de error de validación se deben mantener los valores del formulario ya informados (para no obligar al usuario a introducirlos de nuevo).

Para ello tendremos que informar los atributos `value` de las etiquetas `input` de `nif`, `nombre` y `dirección` con el contenido de las variables php que utilizemos para recuperarlos. Ejemplo:

```
<input type="text" name="nif" id="nif" value='<?=$nif;?>'>
```

Recordad que, para que no se produzca un error al entrar por primera vez en la página, tendremos que inicializar estas variables al inicio del código

NOTA 1: Si queremos validar que una variable contenga un valor podemos utilizar la función `empty($variable)`

NOTA 2: Si, como complemento al ejercicio, queremos que el texto aparezca todo en minúsculas y la primera letra en mayúsculas podemos utilizar:

```
$nombre = ucfirst(strtolower($nombre))
```

- Si no hay errores guardaremos la persona en `$arrayPersonas`. El NIF será la clave del array y `nombre` y `dirección` serán los datos:

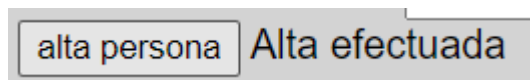
```
$arrayPersonas[$nif]['nombre']
```

```
$arrayPersonas[$nif]['direccion']
```

NOTA: AL utilizar el `nif` que introduce el usuario como clave de la fila del array nos aseguramos que no existan dos filas con la misma clave

- Una vez realizada el alta debería aparecer un mensaje de 'Alta efectuada' en el `span` de

mensajes situado al lado del botón de alta:



- Si todo funciona correctamente añadiremos una validación adicional para que se muestre un error si el NIF que queremos dar de alta ya existe en el array:

Para ello utilizaremos la función `array_key_exists(elemento a buscar, array donde buscarlo)` para comprobar si existe la clave en el array

En caso que exista se mostrará un mensaje de 'persona ya existe'

NOTA: Mientras no habilitemos la operativa de consulta es bastante útil imprimir el array por pantalla para ir viendo su contenido. Podemos utilizar el formato pretty `print_r`

```
echo("<pre>".print_r($arrayPersonas,true)."</pre>")
```

EJERCICIO 4: CONSULTA DE PERSONAS

Al entrar en la página se mostrará en la tabla del documento la lista de personas que se encuentran en la variable `$_SESSION['personas']` y que trasladamos a nuestra variable `$arrayPersonas`

NIF	Nombre	Dirección		
12345678K	Peter clemenza	Piganerelli avenue, 66	Baja	Modificar
22345678J	Virgil solozzo	Foscarelli avenue, 45	Baja	Modificar
32345678P	Luca brasi	Foscarelli avenue, 102	Baja	Modificar
42345678D	Connie corleone	Trattoria street, 10	Baja	Modificar
52345678K	Frank pentangeli	Trattoria street, 101	Baja	Modificar
62345678H	Amerigo bonasera	Margheritti street, 101	Baja	Modificar
72345678Q	Apollonia corleone	Foscarelli avenue, 1	Baja	Modificar

Para realizar esta operativa realizaremos las siguientes acciones:

- Con un bucle `foreach()` recorreremos el array y confeccionamos las filas `<tr>` y columnas `<td>` de la tabla con los datos de cada una de las personas:

```
<tr><td>nif</td><td>nombre y apellidos</td><td>dirección</td><td></td></tr>
```

- Se mostrará el resultado dentro de la etiqueta `<table>` del documento, a continuación de la fila de encabezados

- Al dar de alta una persona nueva se actualizará automáticamente la tabla del documento

con la nueva persona añadida.

OJO porque la ubicación del código para la consulta dentro del flujo general del programa es importante. Primero deberíamos colocar el código que utilizemos para el alta y mantenimiento de personas y, después, el código que lee el array para montar la tabla (de esta forma recogeremos cualquier cambio que hayamos realizado en el array)

- Como complemento del ejercicio mostraremos en la tabla las personas ordenadas por NIF.

Para ello ordenaremos el array de personas utilizando la función `ksort()` (recordad que se utiliza para ordenar un array por sus claves)

EJERCICIO 5: BAJA DE TODAS LAS PERSONAS

Activaremos el botón de 'baja personas', situado bajo la tabla de consulta, para eliminar todas las personas del array

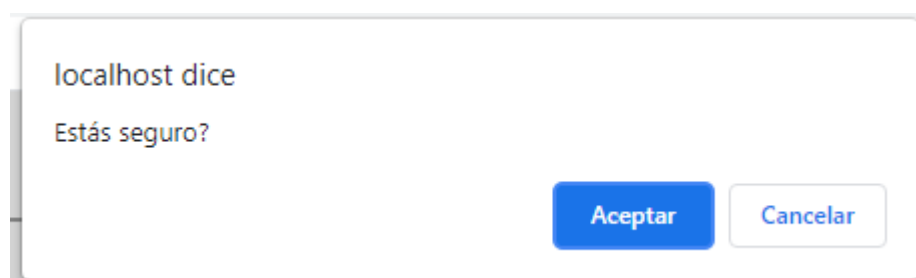
NIF	Nombre	Dirección		
42345678D	Connie corleone	Trattoria street, 10	Baja	Modificar
52345678K	Frank pentangeli	Trattoria street, 101	Baja	Modificar
62345678H	Amerigo bonasera	Margheritti street, 101	Baja	Modificar
72345678Q	Apollonia corleone	Foscarelli avenue, 1	Baja	Modificar

Para ello:

- Tendremos que detectar con la instrucción `isset($_POST['baja'])` cuando el usuario pulsa el botón de 'baja personas'
- Al pulsar el botón volvemos a crear el array `$arrayPersonas` vacío

Ejercicio complementario para los que tengáis conocimientos de JavaScript

Por temas de usabilidad sería recomendable que, al pulsar el botón de baja de personas, preguntemos al usuario para que nos de su confirmación



- Modificaremos el formulario de baja de la siguiente forma:

```
<form method='post' action='#' id='formularioBaja'>
    <input type='hidden' name='baja'></input>
    <input type='button' value='baja personas' id='baja'>
</form>
```

- Con JavaScript activamos un **listener** sobre el botón de baja y le asociaremos una función JS para añadir la confirmación:

```
if (confirm("Estás seguro?")) {
    document.querySelector('#formularioBaja').submit();
}
```

EJERCICIO 6: BAJA INDIVIDUAL DE PERSONAS

Vamos a añadir, al lado de cada persona en la tabla, un botón que permita dar de baja individualmente cada una de las personas que se muestran en pantalla

NIF	Nombre	Dirección		
12345678K	Peter clemenza	Piganerelli avenue, 66	Baja	Modificar
22345678J	Virgil solozzo	Foscarelli avenue, 45	Baja	Modificar

Cuando construimos las filas de la tabla en la consulta de personas añadiremos la siguiente operativa:

- En la tabla, añadimos una cuarta columna con un formulario `<form method='post' action='#'>` que contenga:
 - Un campo de `input` del tipo `hidden` cuyo `value` será el NIF de la persona
 - Un campo de `submit` que será el que utilizaremos para lanzar la baja (recordad asignarle a este botón el atributo `name` correspondiente para poder identificar en php cuando se pulsa sobre el). Ejemplo

```
<input type='submit' name='bajaPersona' value='Baja'>
```

Añadiremos el código php necesario para la operativa de baja:

- Detectar cuando el usuario pulsa el botón de `'bajaPersona'`
- Recuperamos el nif y validamos que llegue informado
- Conviene validar también que el nif a dar de baja existe en el array (con `array_key_exists()`) ya que un usuario avanzado podría modificarlo utilizando la consola del navegador

- Borramos la fila del array utilizando `unset()`:

```
unset($arrayPersonas[$nifBaja]);
```

- Mostramos el mensaje de 'Baja efectuada'

Recordad que, una vez realizada la baja de la persona, la tabla de consulta debe quedar actualizada sin necesidad que el usuario vuelva a refrescar la página.

EJERCICIO 7: MODIFICACIÓN DE PERSONAS

Vamos a añadir, al lado de cada persona en la tabla, un segundo botón que permita modificar individualmente cada una de las personas que se muestran en pantalla.

Además tendremos que transformar en editables las celdas de nombre y apellidos y dirección ya que la modificación la realizaremos directamente en el contenido de las celdas

NIF	Nombre	Dirección		
42345678D	Connie corleone	Trattoria street, 10	Baja	Modificar
52345678K	Frank pentangeli	Trattoria street, 101	Baja	Modificar
62345678H	Amerigo bonasera	Margheritti street, 101	Baja	Modificar
72345678Q	Apollonia corleone	Foscarelli avenue, 1	Baja	Modificar

Modificaciones en la confección de la tabla de consulta y el documento html

Vamos a ver las modificaciones que tendremos que realizar en la consulta de personas cuando montamos las etiquetas de las filas de la tabla:

- Transformar en editables las columnas de la tabla de consulta correspondientes al nombre y dirección. Tenemos dos opciones:

Añadir una etiqueta `<input>` dentro de la celda

```
<td><input type='text' value='{$arrayPersonas[direccion]}></td>
```

Añadir el atributo `contenteditable` en la propia etiqueta `<td>`:

```
<td contenteditable>{$arrayPersonas[nombre]}</td>
```

Connie corleone	Trattoria street, 10
-----------------	----------------------

- Añadiremos un atributo `class` a las celdas de nif, nombre y dirección (si hemos escogido la opción del atributo `contenteditable`) o a la celda nif y los `inputs` de nombre y dirección (si escogemos la opción de etiqueta `<input>`)
- Cuando pulsemos sobre el botón de modificar de la fila correspondiente tendremos, de alguna forma, que enviar los datos modificados al servidor. Para ello confeccionaremos un formulario oculto donde trasladaremos los datos de la fila para que, posteriormente, enviarlos al servidor para modificar la correspondiente fila del array.

```
<form method='post' action='#' id='formularioModi'>
```

```
<input type='hidden' name='nifModi'>
```

```
<input type='hidden' name='nombreModi'>
```

```
<input type='hidden' name="direccionModi">
```

```
<input type='hidden' name='modificar'>
```

```
</form>
```

- En la cuarta columna de la tabla, añadiremos un botón de modificación para cada persona

```
<input type='button' value='Modificar' class='modificar'>
```

NOTA: El botón de ser de tipo `'button'` y no `'submit'` ya que lo utilizaremos solo para asociarle un evento javascript más adelante

	Baja	Modificar	
	Baja	Modificar	
	Baja	Modificar	
	Baja	Modificar	

Activar los botones de modificación con JavaScript

Vamos a activar los botones de modificación para que, cuando se pulsen, se traslade el contenido de

las celdas nif, nombre y dirección al formulario oculto y enviarlos al servidor

- Con JavaScript activaremos un listener para cada uno de los botones

```
//recogemos los botones que tienen la class 'modificar'

var botonesModi = document.querySelectorAll('.modificar');

//recorremos el array con los objetos de tipo botón y, para cada uno de ellos, activamos con
onclick la detección de pulsación del botón. Al pulsar el botón indicamos que se ejecute la
función JS trasladarDatos

botonesModi.forEach(function(item) {

item.onclick = trasladarDatos;

})
```

- Añadimos la función JavaScript `trasladarDatos` que trasladará los datos de la persona modificada en la tabla a los controles `input` del formulario oculto:

- Como primer paso, dentro de la función, tendremos que detectar que botón se ha pulsado (ya que hay uno para cada fila de la tabla).

El selector `this` nos indicará que botón hemos pulsado. Y con `closest()` buscamos la etiqueta `<tr>` padre que corresponde a esa fila

```
let tr = this.closest('tr');
```

- Ahora tenemos que recuperar los valores correspondientes de la persona modificada

```
let nif = tr.querySelector('.nif').innerText;
let nombre = tr.querySelector('.nombre').innerText;
let direccion = tr.querySelector('.direccion').innerText;
```

NOTA 1: con `querySelector()` buscamos, desde la etiqueta `tr`, la etiqueta que tenga la `class` que especifiquemos en el parámetro del método para, posteriormente, con `innerText`, recuperar el contenido de esta etiqueta (que será la celda `td` con el contenido a recuperar)

NOTA 2: Si hemos utilizado etiquetas `<input>` para editar las celdas en vez de `contenteditable` tendremos que utilizar `.value` en lugar de `innerText`

- Trasladamos los valores que hemos recuperado de la tabla al formulario oculto:

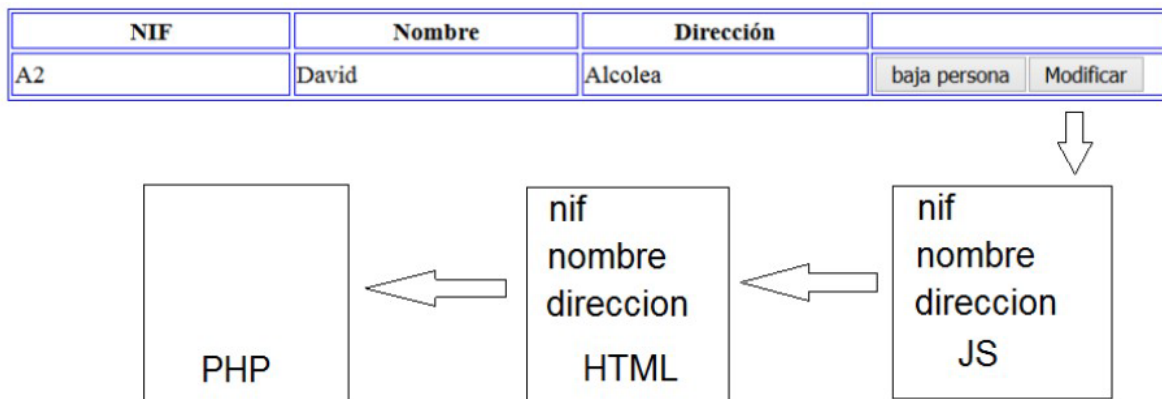
```
document.querySelector('[name=nifModi]').value = nif
document.querySelector('[name=nombreModi]').value = nombre
document.querySelector('[name=direccionModi]').value = direccion
```

NOTA: Con `document.querySelector()` localizamos cada uno de los controles del formulario oculto donde queremos informar el valor obtenido de la celda. Observad como podemos utilizar selectores CSS para localizar cada uno de los controles)

- Por último tenemos que enviar el formulario oculto al servidor desde javascript:

```
document.querySelector('#formularioModi').submit();
```

El esquema de lo que hemos realizado con la operativa anterior sería el siguientes



Modificar en el servidor los datos de la persona

Por último, una vez que enviamos el formulario al servidor, recogeremos los datos con php para modificar la persona en el array de personas:

- Detectar cuando nos llega el formulario de modificación (podemos preguntar por `$_POST['modificar']` que es uno de los inputs de formulario)
- Recuperamos los datos y realizamos las mismas validaciones de obligatoriedad que en el alta (nif, nombre y dirección)

NOTA 1: Puesto que las validaciones son las mismas en alta y modificación podemos incluir éstas en una función para evitar repetir código

NOTA 2: Si hemos utilizado etiquetas `<input>` en las celdas de edición nos bastará utilizar `trim()` para suprimir los espacios en blanco, pero si hemos utilizado el atributo `contenteditable` tendremos que suprimirlos utilizando `filter_input` con filtro de saneamiento:

```
trim(filter_input(INPUT_POST, 'nombre', FILTER_SANITIZE_SPECIAL_CHARS, FILTER_FLAG_STRIP_HIGH))
```

Este filtro escapa caracteres HTML "'<>&" y caracteres con valores ASCII menores que 32 (los espacios en blanco del innerText de la etiqueta `<td>` vienen codificados con la entidad html ` `)

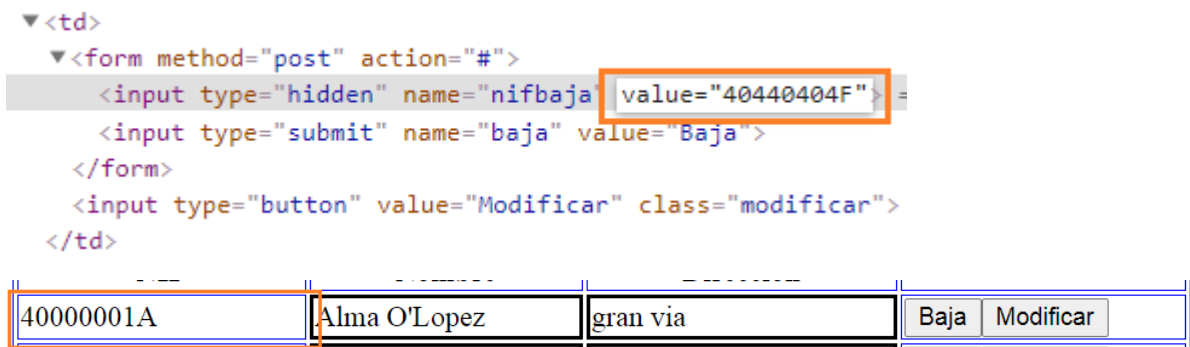
- Validamos, a diferencia que en el alta, que el NIF exista en el array (podemos utilizar la función php `array_key_exists(clave a buscar, array donde buscar)`)
- Modificamos la fila en el array

```
$arrayPersonas[$nif]['nombre'] = $nombre;
$arrayPersonas[$nif]['direccion'] = $direccion;
```
- Una vez actualizada se refrescará el contenido de la tabla del documento y se mostrará un mensaje de 'modificación efectuada'

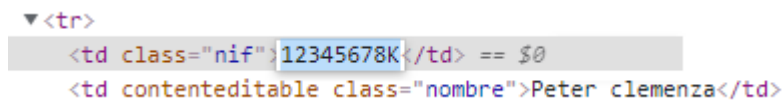
EJERCICIO 8 COMPLEMENTARIO: PROTEGER NIF EN BAJA Y MODIFICACIÓN

Como desarrolladores tenemos que pensar en cualquier casuística que se derive de usar un FrontEnd web y, por tanto, de código abierto.

Un usuario avanzado podría modificarnos el NIF asociado a la baja y a la modificación utilizando la consola. Por ejemplo, podríamos modificar el nif de cualquiera de las filas en el `input hidden` asociado al botón de baja de cada persona y asignar un nif distinto que no corresponda con la fila a dar de baja:



También podríamos modificar por consola el nif visible en la tabla y que es necesario para realizar la modificación de datos:



Tendremos un problema si el NIF substituido no existe y un problema mucho mayor si éste pertenece a una persona distinta.

Para dificultar la manipulación vamos a utilizar un sistema de cifrado del NIF para que, en caso de que el usuario lo modifique, sea casi imposible que sea un NIF válido. Para ello utilizaremos una máscara XOR:

1. Creamos en una constante una clave de cifrado cualquiera. Por ejemplo:
`const CLAVE = 'zxspectrum'`
2. Cuando creamos las filas de la tabla ciframos el NIF asociado a la baja y modificación utilizando una máscara XOR (^):
`$nifCifrado = $nifArray ^ CLAVE;`
3. Informamos el atributo value del input hidden de la baja con el nif encriptado:
`<input type='hidden' name='nif' value='$nifCifrado'>`
4. En el caso de la modificación vamos a crearnos en la etiqueta <td> correspondiente al nif un atributo específico para guardar el nif cifrado (es obvio que el usuario tiene que visualizar el nif sin cifrar)
`<td class='nif' data-nif = '$nifCifrado'>$nifArray</td>`

5. Cuando el usuario pulse el botón de baja y recuperemos el NIF cifrado, en el servidor tendremos que realizar el proceso inverso. Para ello solo tenemos que volver a utilizar la misma clave para descifrarlo:

```
$nifBaja = $_POST['nif del formulario de baja'] ^ CLAVE;
```

6. Si pulsamos sobre el botón de modificación, tendremos que enviar el nif cifrado al servidor y no el nif visible en la tabla. Para ello tendremos que modificar el script de recuperación de datos de la fila para cambiar el origen del dato (en vez de recuperar el contenido de la celda td asociada al nif recuperaremos el contenido del atributo data-nif)

```
let nif = tr.querySelector('.nif').getAttribute('data-nif');
```